www.software.ac.uk

FAIR Software?

How can we make easier to find, access, deposit and reuse software?

Slides: https://doi.org/10.6084/m9.figshare.5620690

22nd November 2017, Dealing With Data 2017, Edinburgh

Neil Chue Hong (@npch), Software Sustainability Institute
ORCID: 0000-0002-8876-7606 | N.ChueHong@software.ac.uk

Supported by BBSRC EPSRC Pioneering research E·S·R·C Economic & Social Research Council

Project funding from Jisc NERC Science of the Environment

Slides: https://doi.org/10.6084/m9.figshare.5620690

www.software.ac.uk

Findable
Accessible
Interoperable
Reusable

Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data 3:160018 doi: 10.1038/sdata.2016.18 (2016).

Slides: https://doi.org/10.6084/m9.figshare.5620690

www.software.ac.uk

"FAIRness is a prerequisite for proper data management and data stewardship"

What does this mean when applied to software?

Research Software Workflow

www.software.ac.uk

describe

GitLab

GitHub

Bitbucket

<sword />

zenodo

DRYAD

figshare
credit for all your research

develop → share → preserve

Developed and versioned using code repository

Published via code repository or website

Deposited in digital repository with paper / for preservation

Findable

assigned unique identifier

described with rich metadata

indexed in searchable resource

First: Use a code repository

www.software.ac.uk

- Makes software management *much* easier
- Will help make software FAIR (as you'll see)

GitLab

https://education.github.com/discount_requests/new

Identifying software

www.software.ac.uk

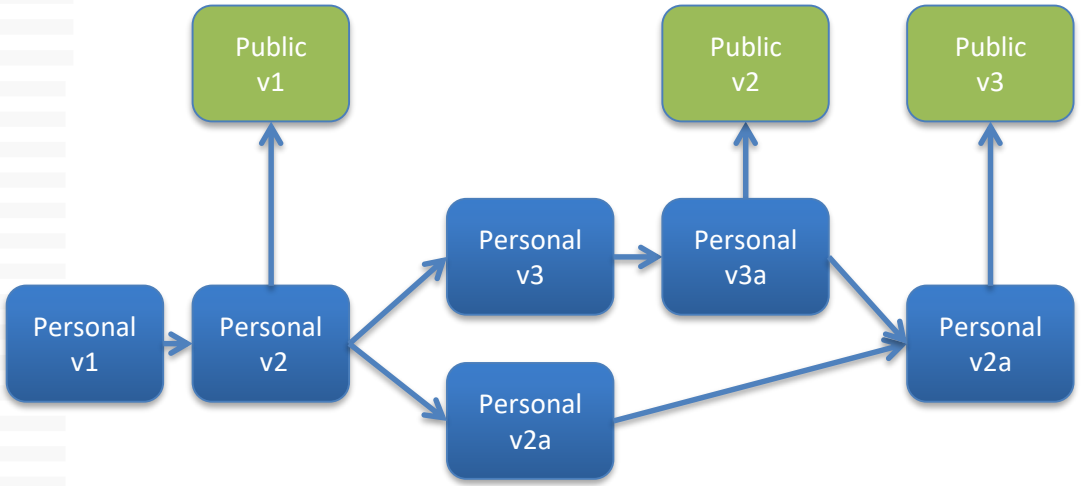
Program

Function

Algorithm

```c
#include <stdio.h>
#include <stdlib.h>

void swap(int *x,int *y)
{
   int temp;
   temp = *x;
   *x = *y;
   *y = temp;
}

int choose_pivot(int i,int j )
{
   return((i+j) /2);
}

void quicksort(int list[],int m,int n)
{
   int key,i,j,k;
   if( m < n)
   {
      k = choose_pivot(m,n);
      swap(&list[m],&list[k]);
      key = list[m];
      i = m+1;
      j = n;
      while(i <= j)
      {
         while((i <= n) && (list[i] <= key))
                i++;
         while((j >= m) && (list[j] > key))
                j--;
         if( i < j)
                swap(&list[i],&list[j]);
      }
      // swap two elements
      swap(&list[m],&list[j]);
      // recursively sort the lesser list
      quicksort(list,m,j-1);
      quicksort(list,j+1,n);
   }
}
void printlist(int list[],int n)
{
   int i;
   for(i=0;i<n;i++)
      printf("%d\t",list[i]);
}

void main()
{
   const int MAX_ELEMENTS = 10;
   int list[MAX_ELEMENTS];

   int i = 0;

   // generate random numbers and fill them to the list
   for(i = 0; i < MAX_ELEMENTS; i++ ){
       list[i] = rand();
   }
   printf("The list before sorting is:\n");
   printlist(list,MAX_ELEMENTS);

   // sort the list using quicksort
   quicksort(list,0,MAX_ELEMENTS-1);

   // print the result
   printf("The list after sorting using quicksort algorithm:\n");
   printlist(list,MAX_ELEMENTS);
}
```

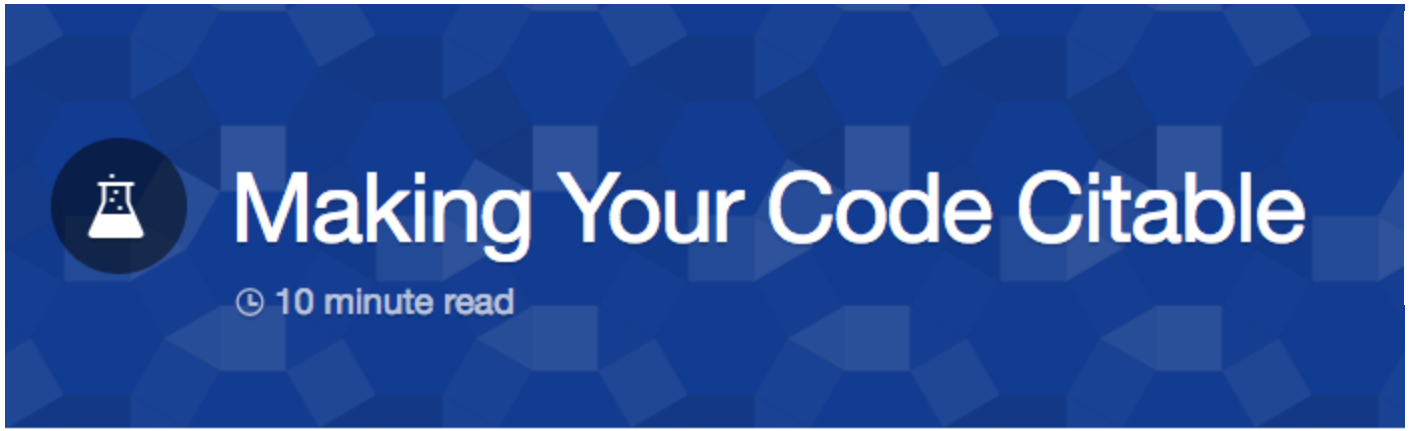
- Identifying software is hard
- Giving software an identifier is easy



Sustainability Institute

Giving software an identifier

www.software.ac.uk



GitHub

Bitbucket

zenodo

figshare
credit for all your research

<sword />

Digital Object Identifiers (DOI) are the backbone of the academic reference and metrics system. If you're a researcher writing software, this guide will show you how to make the work you share on GitHub citable by archiving one of your GitHub repositories and assigning a DOI with the data archiving tool Zenodo.

> **ProTip:** This tutorial is aimed at researchers who want to cite GitHub repositories in academic literature. Provided you've already set up a GitHub repository, this tutorial can be completed without installing any special software. If you haven't yet created a project on GitHub, start first by uploading your work to a repository.

42,000 software packages with DOIs (2013-2017)

Rich metadata is the issue

- Many programmers don't like to describe their software
- Not always clear what should be described to make things findable
- However:
  - A lot of metadata can be automatically generated
  - Start small and improve
- A README file is the human-readable starting point

CodeMeta: A Rosetta Stone for Software Metadata

www.software.ac.uk

The CodeMeta Project

Crosswalk for GitHub API

Property	GitHub
codeRepository	html_url
programmingLanguage	languages_url
downloadUrl	archive_url
author	login
dateCreated	created_at
dateModified	updated_at
license	license
description	description
identifier	id
name	full_name
issueTracker	issues_url

The CodeMeta Project

Crosswalk for Zenodo metadata

Zenodo.org is a data archive based at CERN which is popularly used to archive and provide DOIs to academic software from GitHub, as described in the official GitHub guide to Making your code citable.

Property	Zenodo
codeRepository	relatedLink
applicationCategory	communities
author	creators
datePublished	date_published
funder	contributors.Funder
keywords	keywords
license	license
description	description/notes
identifier	id
name	title
affiliation	affiliation
identifier	ORCID
name	name

Matthew B. Jones et al. 2017. **CodeMeta: an exchange schema for software metadata. V2.0.** KNB Data Repository. doi:10.5063/schema/codemeta-2.0

Software Sus

Accessible

www.software.ac.uk

retrievable using standard, open, free protocol

metadata accessible even when software is no longer available

Software is easy to make accessible

- By using code repositories (for development) and digital repositories (for preservation) software is accessible
- By using a digital repository which issues DOIs, metadata is guaranteed to be kept available
- But… is this metadata useful?
  - It is for citation, which improves reuse and credit
    - Software Citation Principles. PeerJ Computer Science 2:e86. DOI: 10.7717/peerj-cs.86
  - ORCIDS make it easier to tie together

Interoperable

formal, accessible, shared language for knowledge representation

qualified references to other metadata

CodeMeta built on Schema.org and JSON-LD


```
{
  "@context": [
    "https://doi.org/10.5063/schema/codemeta-2.0",
    "http://schema.org"
  ],
  "@id": "https://doi.org10.5281/zenodo.1048320",
  "@type": "SoftwareSourceCode",
  "identifier": "codemetar",
  "description": "The 'Codemeta' Project defines a 'JSON-LD' format for
describing\n  software metadata, as detailed at
<https://codemeta.github.io>. This package\n  provides utilities to
generate, parse, and modify 'codemeta.jsonld' files \n  automatically for
R packages, as well as tools and examples for working with\n  'codemeta'
'JSON-LD' more generally.",
  "name": "codemetar: Generate CodeMeta Metadata for R Packages",
  "issueTracker": "https://github.com/codemeta/codemetar/issues",
  "license": "https://spdx.org/licenses/MIT",
  "version": "0.1.2",
  "programmingLanguage": {
    "@type": "ComputerLanguage",
    "name": "R",
    "version": "3.4.2",
    "url": "https://r-project.org"
  },
  "runtimePlatform": "R version 3.4.2 (2017-09-28)",
  "provider": {
    "@id": "https://cran.r-project.org",
    "@type": "Organization",
    "name": "Central R Archive Network (CRAN)",
    "url": "https://cran.r-project.org"
  },
  "author": [
    {
      "@type": "Person",
      "givenName": "Carl",
      "familyName": "Boettiger",
      "email": "cboettig@gmail.com",
      "@id": "http://orcid.org/0000-0002-1642-628X"
    }
  ],
    {
      "@type": "SoftwareApplication",
      "identifier": "stringi",
      "name": "stringi",
      "provider": {
        "@id": "https://cran.r-project.org",
        "@type": "Organization",
        "name": "Central R Archive Network (CRAN)",
        "url": "https://cran.r-project.org"
      }
    },
    {
      "@type": "SoftwareApplication",
      "identifier": "readr",
      "name": "readr",
      "provider": {
        "@id": "https://cran.r-project.org",
        "@type": "Organization",
        "name": "Central R Archive Network (CRAN)",
        "url": "https://cran.r-project.org"
      }
    },
    {
      "@type": "SoftwareApplication",
      "identifier": "R",
      "name": "R",
      "version": "3.0.0"
    }
  ],
  "codeRepository": "https://github.com/ropensci/codemetar",
  "isPartOf": "https://ropensci.org",
  "keywords": ["metadata", "codemeta", "ropensci", "citation", "credit",
"linked-data"],
  "relatedLink": "https://codemeta.github.io/codemetar",
  "contIntegration": "https://travis-ci.org/ropensci/codemetar",
  "developmentStatus": "active",
  "releaseNotes":
"https://github.com/ropensci/codemetar/blob/master/NEWS.md",
  "readme": "https://github.com/ropensci/codemetar/blob/master/README.md",
  "fileSize": "662.402KB"
}
```

Reusable

clear and accessible license

detailed provenance

meet community standards

Clarity is quite complicated

www.software.ac.uk

- Software Package Data Exchange (SPDX) increasingly used for licensing information
  - But all you need to know is that you should choose a commercial license or one from this list:
    - https://opensource.org/licenses
- Provenance is simple (from code repository) and difficult (from underlying research)
  - Jupyter Notebooks can make things easier
  - Principle of "recording your working" – see next talk!

Community standards depend on the community!

- ESIP (Earth Sciences): https://esipfed.github.io/Software-Assessment-Guidelines/
- CLARIAH (Arts and Humanities): https://github.com/CLARIAH/software-quality-guidelines
- IPOL (Image Processing): https://tools.ipol.im/wiki/ref/software_guidelines/
- ELIXIR (Life Sciences): https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5490478/

Research Software Workflow

www.software.ac.uk

describe

GitLab

GitHub

Bitbucket

<sword />

zenodo

DRYAD

figshare
credit for all your research

develop → share → preserve

Developed and versioned using code repository

Published via code repository or website

Deposited in digital repository with paper / for preservation

Making Software FAIR

- Get an ORCID for yourself
- Develop your software in a code repository
- Choose a clear license for your software
- Keep essential metadata and README in your code repository up to date
  - Codemeta.json file in future
- Archive major versions, to get a DOI
- Cite your software in your papers, and put your preferred citation in your README

Interested in doing more?

- FORCE11 Software Citation Implementation Working Group
  - https://www.force11.org/group/software-citation-implementation-working-group
- RDA Software Source Code Interest Group
  - https://www.rd-alliance.org/groups/software-source-code-ig
- SWORD v3 development
  - http://swordapp.org/
- Software Sustainability Institute
  - https://www.software.ac.uk/

Find out more about the SSI

www.software.ac.uk

- Community Engagement (Lead: Shoaib Sufi)
  - Fellowship Programme
  - Events and Workshops
- Consultancy (Lead: Steve Crouch)
  - Open Call for Projects / Collaborations
  - Software Evaluation
- Policy and Publicity (Lead: Simon Hettrick)
  - Case Studies / Policy Campaigns
  - Software and Research Blog
- Training (Lead: Aleksandra Nenadic)
  - Software Carpentry / Data Carpentry (300+ students/year)
  - Guides and Top Tips
- Journal of Open Research Software (Editor: Neil Chue Hong)

- Collaboration between universities of Edinburgh, Manchester, Oxford and Southampton
Supported by EPSRC Grant EP/H043160/1 + EPSRC/ESRC/BBSRC grant EP/N006410/1